WWW.SCIAM-MAGAZINE.NAROD.RU

Графотворчество невидимого профессора, скрытого за экраном дисплея

А. К. ДЬЮДНИ

Если уподобить экран компьютерного дисплея классной доске, то кто же тогда рисует на ней красивые плавные кривые? Наверное, невидимый профессор — электронное воплощение преподавателей аналитической геометрии. Некоторые старательно выводимые ими кривые носят экзотические названия, такие как локон Аньези, конхоида Никомеда и улитка Паскаля. Среди них встречаются параболы, циклоиды, спирали и синусоиды. Они едва не были преданы забвению, когда в математике стали проявляться тенденции к обобщениям, а геометрия стала не евклидовой. Теперь значение этих кривых в основном заключается в отражении формы, которую принимают физические функции. Кривые напоминают нам о чудесных открытиях — и знаменитых, и не очень известных, сделанных математиками прошлых столетий.

Невидимый профессор — это набор простейших компьютерных программ. При изображении он передает красоту кривых с такой выразительностью, о которой едва ли могли мечтать их первооткрыватели. Мел в руке профессора — это маленькая яркая точка на экране. Изображение графика кубической функции Чирнгаузена, например, начинается с правого верхнего угла; кривая плавно проходит через начало координат, делает петлю и снова возвращается в начало координат. Далее светящаяся точка «прожигает» свой путь к нижнему правому углу экрана, где останавливается и гаснет (см. левую часть рисунка внизу).

Невидимый профессор может нарисовать даже целое семейство таких кривых. Кубическое уравнение Чирнгаузена содержит постоянную величину, или параметр. Задавая различные значения этого параметра, мы получаем различные версии кривой — на экране как бы воплощаются вариации на заданную тему. Недостает только старческого голоса: «Ну, а здесь с первого взгляда на формулу мы видим, что это обобщенная полукубическая парабола».

Для невидимого профессора не составляет труда изобразить трансцендентный пейзаж из волнистых «параллельных» кривых или «комплексный» океан, искривляя прямоугольную форму ячеек решетки. Таких программ любители геометрии написали уже немало.

Чтобы дать читателям минимальные сведения о том, как рисовать различные кривые, я сам на некоторое время превращусь в невидимого профессора. В качестве примера рассмотрим, как составить программу, чтобы на экране получить обычную окружность. Для этого вспомним три способа задания окружности.

Первый и наиболее привычный способ знаком нам со школы:

$$x^2 + y^2 = c^2.$$

Это уравнение множества точек, принадлежащих окружности. Любая точка с декартовыми координатами x и y лежит на окружности тогда и только тогда, когда сумма квадратов x и y равна квадрату c (c — радиус окружности). Каждый раз при выполнении программы пользователь должен указать значение c.

Приведенной выше формулой можно воспользоваться для составления программы, рисующей окружность, если разделить кривую на две части — верхнюю и нижнюю. Программа пробегает через последовательные значения x, высвечивая на экране соответствующие им точки y. Чтобы строить изображение точки, значение y должно быть вычислено без квадрата, поэтому уравнение нужно решить относительно y. Две ветви решения отличаются только знаком:

$$y = +\sqrt{c^2 - x^2},$$

$$y = -\sqrt{c^2 - x^2}.$$

Однако такое задание кривой выглядит уже несколько неуклюже. Наверное, лучше воспользоваться полярными координатами.

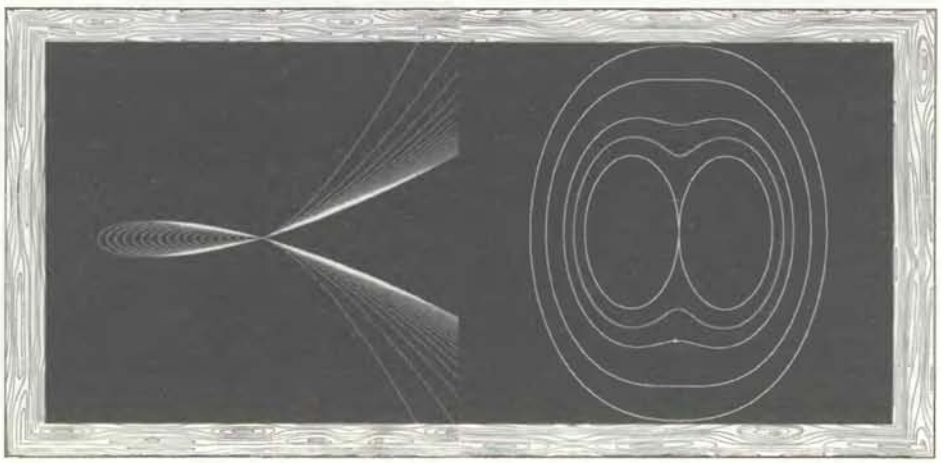
В декартовой системе координаты x и y указывают удаление данной точки соответственно по горизонтали и по вертикали от центральной точки, называемой началом отсчета. Полярные координаты r и θ определяют точку совершенно по-другому: r — это расстояние, на котором данная точка находится от начала отсчета, а θ — угол между линией, соединяющей данную точку с точкой начала отсчета, и горизонтальной прямой. Уравнение окружности в полярных координатах имеет очень простой вид: $r = c$. Независимо от величины угла θ расстояние r всегда равно c. К сожалению, лишь немногие читатели могут воспользоваться языками программирования, в которых для вывода изображений можно применять полярные координаты.

Третье представление основано на так называемых параметрических уравнениях. Для многих программ, рисующих изображения кривых, это представление наиболее подходящее:

$$x = c \cdot \cos(t),$$

$$y = c \cdot \sin(t).$$

Здесь параметр t пробегает значения от 0 до 360°. Можно высвечивать точки (x, y) с интервалом, скажем, в 1°. Читатели, немного подзабывшие функции синуса и косинуса, в крайнем



Двадцать версий кубической параболы Чирнгаузена (слева) и пять версий гиппопеды (справа)

WWW.SCIAM-MAGAZINE.NAROD.RU

случае могут прочесть о них в своем руководстве по языку программирования.

На алгоритмическом уровне «скелет» программы, которую я назову CIRCLE (круг), мог бы выглядеть следующим образом:

```
input c
for t ← 1 to 360
    x ← c · cos(t)
    y ← c · sin(t)
    точка (x, y)
```

Текст программного фрагмента не будет значительно отличаться от этого алгоритма. В конце фрагмента будет стоять та или иная графическая команда, роль которой здесь играет «точка (x, y)».

Однако реальная программа, рисующая окружность, конечно, не будет такой простой. Во-первых, у большинства персональных компьютеров начало координат находится в углу экрана. Следовательно, нужно добавить смещение, чтобы переместить окружность в центральную часть экрана. Если, например, на экране можно разместить 200 точек по горизонтали и 150 точек по вертикали, то приведенный выше программный фрагмент нужно модифицировать следующим образом:

$$x \leftarrow c \cdot \cos(t) + 100$$

$$y \leftarrow c \cdot \sin(t) + 75$$

Теперь невидимый профессор может воплотиться в нескольких маленьких программках, в основе которых лежит та же самая структура. Меняться будут лишь уравнения. Например, кубическая кривая Чирнгаузена имеет параметрические уравнения, в которых нет ни синуса, ни косинуса:

$$x = 3a(t^2 - 3),$$

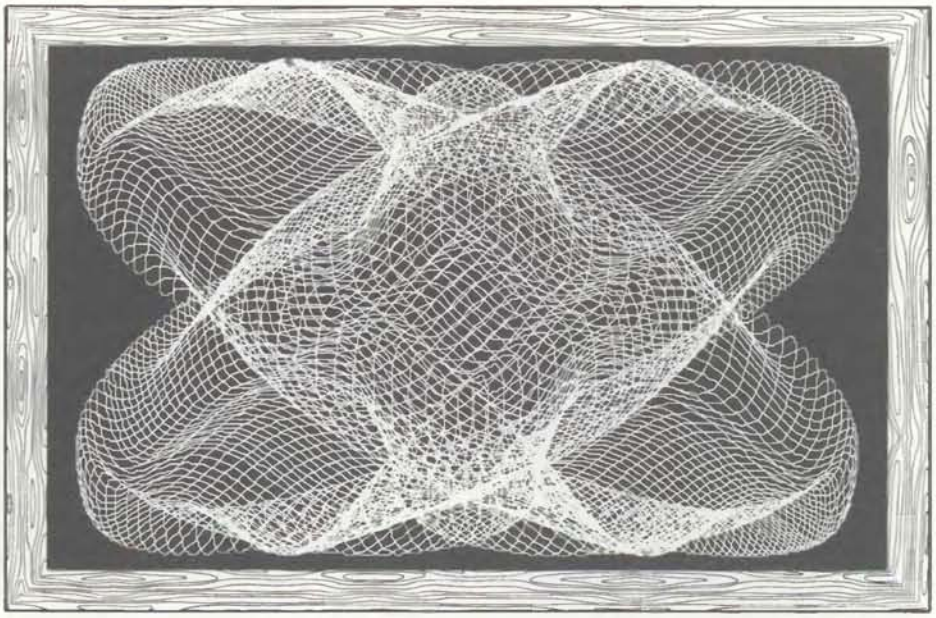
$$y = at(t^2 - 3).$$

Поэтому в программе, которую я назвал TSCHIRNHAUSEN, используется параметр t, пробегающий от минимального до максимального значения. Эти значения определяются размерами экрана. Например, на упомянутом выше экране размером 150 x 200 точек поместится кривая со значениями t в диапазоне от $t = -4{,}4$ до $t = +4{,}4$, если $a = 1$. Ни один цифровой компьютер не может варьировать параметр t в этом диапазоне непрерывным образом — каждый раз величина t должна получать определенное приращение, достаточно малое, чтобы создать впечатление непрерывности. Другими словами, мел не должен крошиться при нажиме, чтобы линия получилась плавной. Поскольку наша воображаемая доска имеет 200 точек по горизонтали, разумное приращение для параметра t можно получить делением диапазона возможных его значений (а именно 8,8) на 200 равных частей. Таким образом получаем величину приращения, равную 0,044. В то же время следует учесть, что кривая рисуется не везде равномерно. Некоторые участки «мел» проходит быстрее, и на них точки изображения отстоят несколько дальше друг от друга. К счастью, подобные вычисления обходятся настолько дешево (и выполняются так быстро), что размер шага можно сократить до 0,01. Теперь наш алгоритм будет выглядеть следующим образом:

```
input a
for t ← −4.4 to + 4.4 step 0.01
        x ← 3 a (t² − 3) + 100
        y ← at (t² − 3) + 75
        точка (x, y)
```

Другой простой способ сделать кривую более гладкой заключается в том, чтобы воспользоваться отрезками прямых линий. Программу TSCHIRNHAUSEN так же, как и любую другую из описанных здесь программ, можно модифицировать таким образом, чтобы каждая новая точка получалась в результате выполнения команды, рисующей отрезок прямой, соединяющей предыдущую точку с новой. В этом случае координаты старой точки каждый раз должны заменяться координатами новой. Если отрезки достаточно малы, то человеческий глаз не отличит полученную таким образом ломаную линию от плавной кривой.

В любой версии программы следует также добавить новый внешний цикл, в котором константа a в диапазоне от 0,1 до 2 на каждом шаге принимает приращение, равное 0,1. В этом случае невидимый профессор нарисует целое семейство красиво вложенных друг в друга кубических кривых, которыми сам Чирнгаузен мог бы только гордиться.



Кривая С. Миллера «Безумие» — наглядный пример случайно сделанных открытий

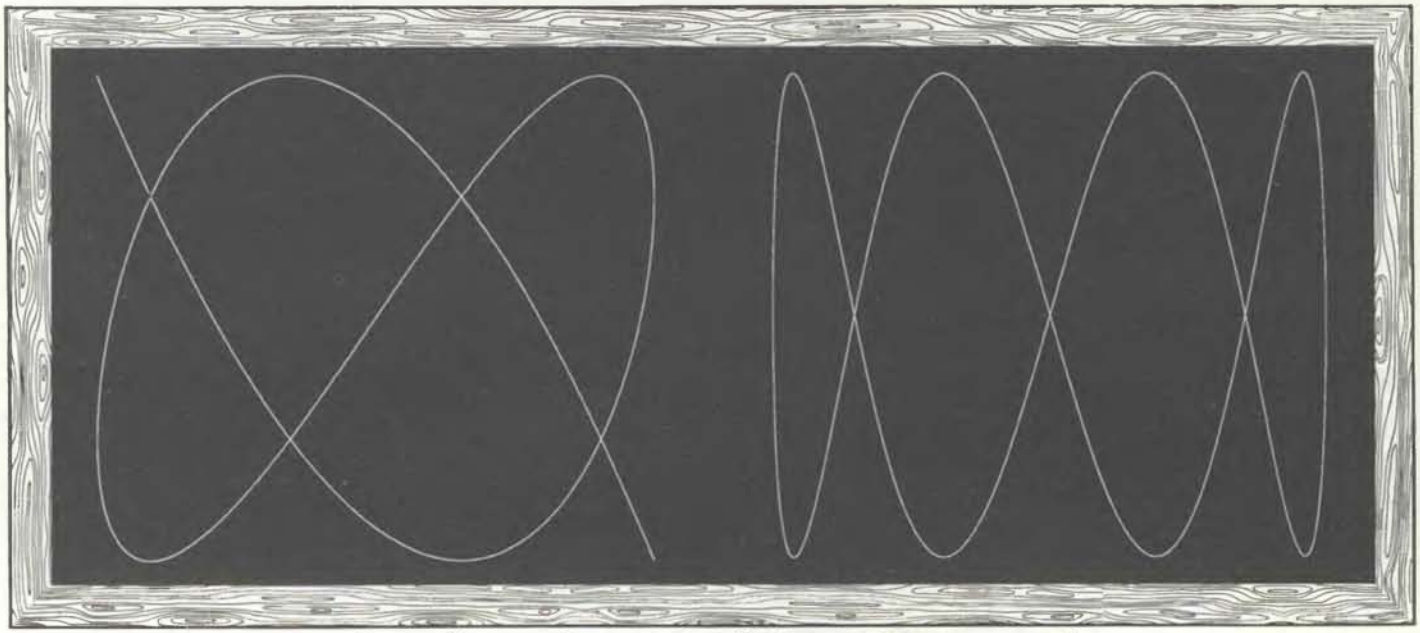
Гиппопеда, изучавшаяся еще греческим философом Проклом приблизительно в 475 г., может принимать две формы — либо в виде лежащей на боку восьмерки, либо в виде гантели в зависимости от значений ее двух констант a и b (см. правую часть рисунка на с. 80). Параметрические уравнения этой кривой ненамного сложнее предыдущих:

$$x = 2\cos(t)\sqrt{ab - b^2\sin^2 t},$$

$$y = 2\sin(t)\sqrt{ab - b^2\sin^2 t}.$$

Параметр t пробегает значения от $-180°$ до $+180°$. Здесь время на вычислительную работу можно несколько сэкономить, поскольку выражение под знаком квадратного корня присутствует в обоих уравнениях. Таким образом программа, названная мной HIPPOPEDE, вычисляет это выражение лишь один раз, записывает ее в качестве значения переменной *temp*, а затем умножает переменную *temp* на $2\cos(t)$ и на $2\sin(t)$. Здесь читатели, которые пожелают создать своего собственного невидимого профессора, вновь сталкиваются с проблемой выбора шага. Хотя размер шага в какой-то степени зависит от параметров a и b, размера экрана и некоторых других факторов, шаг величиной в 1° в большинстве случаев должен оказаться подходящим. Опять-таки следует попробовать различные значения констант a и b, либо целые, либо дробные в диапазоне значений, скажем, от 1 до 10. В результа-

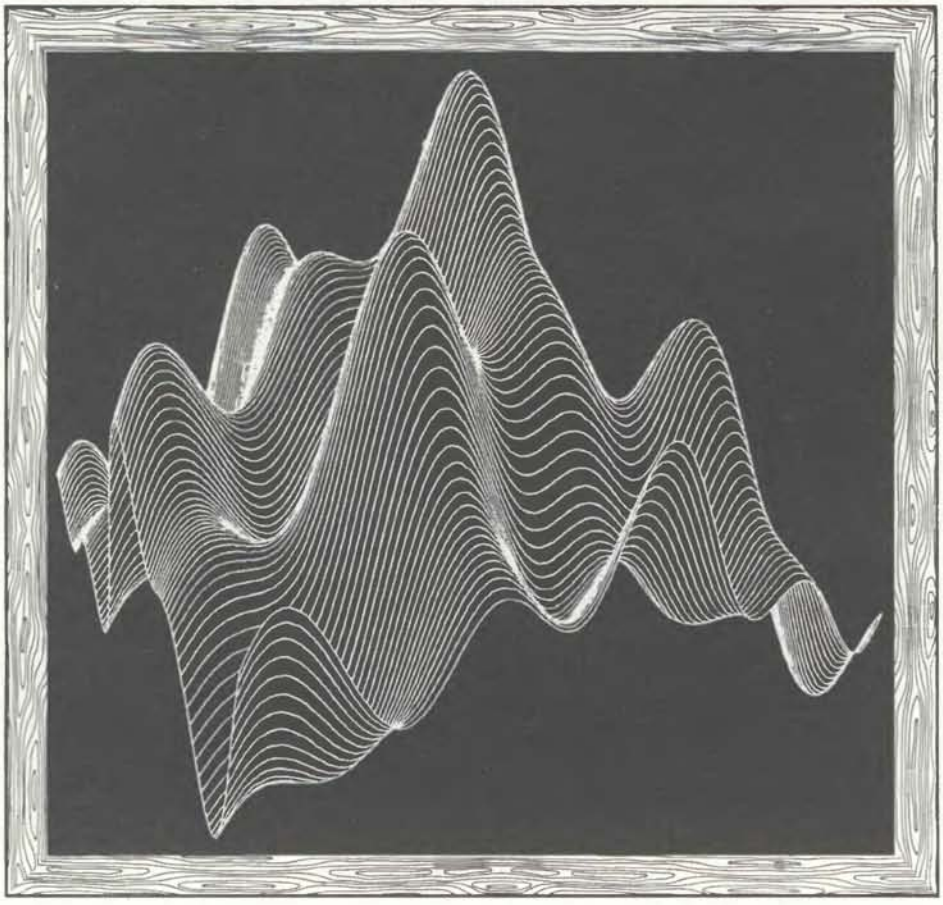
WWW.SCIAM-MAGAZINE.NAROD.RU



Фигуры Лиссажу, одна из которых бесконечно повторяет саму себя

те должно получиться изображение «старой мудрой совы», если для каждой кривой значения a выбрать равными 20, 25, 30, 40 и 50, а b поддерживать равным 20.

Спираль, как одна из красивейших форм, встречающихся в природе, является излюбленной темой невидимого профессора. Неудивительно, что их изображение лучше всего строить в системе полярных координат. Например, любая точка, принадлежащая спирали Архимеда и имеющая полярные координаты r и θ, удовлетворяет тривиальному, как говорят математики, уравнению $r = a\theta$. Значение угла θ растет от 0 до тех пор, пока позволяют размеры экрана (а также константа a). В качестве приращения можно взять 1 или еще меньшую величину. Если значение a принять равным 0,01, то спираль будет закручена довольно плотно, при $a = 0{,}1$ она получается уже намного более «разреженной».



Сложная поверхность из простых функций

Невидимый профессор может без труда справиться и со знаменитой логарифмической спиралью. Эту кривую можно видеть в морских ракушках и на головках подсолнуха. И в этом случае невидимый профессор пользуется полярными координатами, в которых уравнение спирали имеет вид $r = e^{a\theta}$, где иррациональное число e можно принять равным 2,7183. Размер шага здесь можно выбрать таким же, как и для спирали Архимеда, однако читателей следует предупредить, что экспоненциальные функции растут с чудовищной скоростью: даже при $a = 0{,}01$ спираль раскручивается очень быстро.

Описания не менее 300 интересных кривых содержатся в книжке небольшого формата Дж. Денниса Лоуренса «Каталог плоских кривых» (J. D. Lawrence, A Catalog of Special Plane Curves). Здесь невидимому профессору, так сказать, есть где разгуляться. Более опытным программистам, возможно, будет интересно попытаться собрать все программки, рисующие кривые, в одну большую программу. При этом к частным подпрограммам можно было бы обращаться, совершая выбор в так называемом меню.

В некоторых случаях не нужно даже собирать несколько программ в единое целое. Искусство обобщения позволяет создать одну универсальную программу, которая генерирует все кривые данного класса. Пользователю остается только задать значения различных констант в основной исходной формуле. Например, окружность — это лишь одна из множества кривых, называемых фигурами Лис-

WWW.SCIAM-MAGAZINE.NAROD.RU

сажу и описываемых двумя параметрическими уравнениями:

$$x = a \cdot \sin(ct + d),$$
$$y = b \cdot \sin(t).$$

Эти кривые хорошо знакомы тем, кто работает с осциллографом. На экране осциллографа они порождаются электронными сигналами, изменяющимися по синусоидальному закону.

Константы a и b определяют прямоугольник длиной a и высотой b единиц. В пределах этого прямоугольника фигуры Лиссажу исполняют свой неистовый танец. Константы c и d определяют характер этого танца и форму кривых. Например, когда $d = 0$, значения $c = 3/5$ и $c = 1/4$ порождают две кривые, показанные на рисунке вверху на с. 82. Если значение c рационально, то кривая возвращается к своей исходной точке и повторяется до тех пор, пока невидимый профессор не исчерпает всех своих сил. Не стоит беспокоиться о том, рациональны или иррациональны числа в вашем компьютере: цифровой компьютер работает только с рациональными числами. (В предыдущих рассуждениях мы предполагали, что t измеряется в радианах. Чтобы перевести уравнения в градусную меру, нужно умножить аргумент функции sin на число $\pi/180$.) Если исходное значение t равно 0, то фигура Лиссажу начинается где-то на оси x. Вообще говоря, она совершает различные замысловатые траектории, прежде чем снова замкнется. Однако насколько большим будет значение t, когда это наконец произойдет? Читателям, любящим поразмышлять, предлагается самим ответить на этот вопрос. Пусть, например, t не превосходит значения 3600, изменяется с шагом 1 и принимает рациональные дробные значения с не очень большим количеством цифр в числителе и знаменателе. Ну, а любопытным, юмористам или просто страстным любителям головоломок можно предложить задачу с неизвестным исходом. Пусть они попытаются исчерпать возможности невидимого профессора, не устанавливая верхнего предела для t, и определить, сколько времени потребуется на то, чтобы кривая замкнулась. В этом случае цикл в программе строится чисто механически. В конце цикла обычный оператор «go to» передает управление в программе на начало цикла. Чтобы не приковать навечно старика-учителя к доске, вставим в цикл, из милосердия, один условный оператор «if», так что если будет нажата клавиша, то программа выйдет на определенный оператор за пределами цикла.

Среди факторов, побудивших заняться анализом графических возможностей компьютера, было письмо от С. Миллера из Конкорда (шт. Массачусетс). Миллер, по долгу службы консультирующий менеджеров, увлекся таинственными кривыми, являющимися графическим выражением различных функций. Все началось с коммерческой программы CURVES, о которой далее я скажу еще несколько слов.

«Как-то раз поздно вечером, после напряженной работы, — пишет Миллер, — я случайно нажал не на ту клавишу, и передо мной возникла кривая, совершенно не похожая на те циклоиды, которые я генерировал до сих пор... Сначала я подумал, что в компьютере попало немного сыра с моего сандвича, который я жевал, сидя перед дисплеем, но затем я снова запустил программу с другими параметрами (константами) и вновь получил почти то же самое». Это неожиданное событие навело Миллера на мысль о том, что достаточно универсальная программа, генерирующая кривые, была бы способна порождать весьма разнообразные и весьма занятные изображения. Одну из его находок я поместил на с. 81. Она выглядит как трехмерная сеть, закинутая в поисках впечатлительных умов. Если эта кривая нуждается в названии, то я бы назвал ее «Безумием». Ее параметрические уравнения на вид кажутся очень простыми:

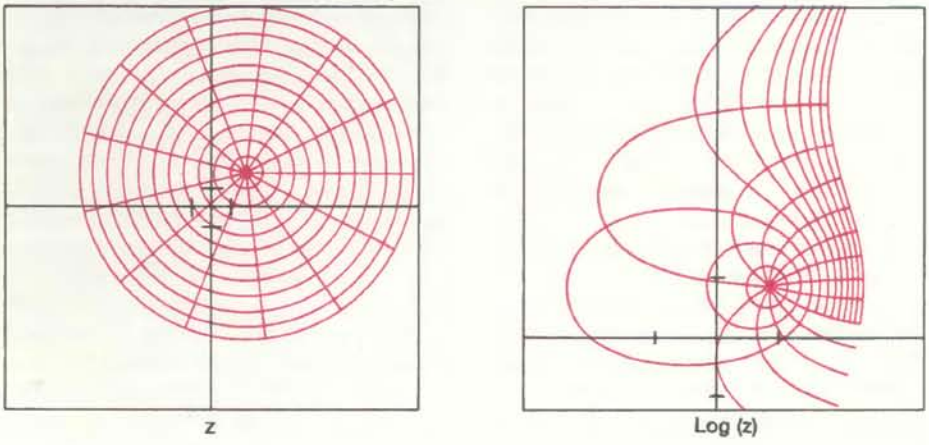
$$x = \sin(0{,}99\,t) - 0{,}7\cos(3{,}01\,t),$$
$$y = \cos(1{,}01\,t) + 0{,}1\sin(15{,}03\,t).$$

Я понятия не имею, насколько велико должно быть значение t, чтобы завершить изображение. Читателям, которые пожелают увидеть «Безумие» на своих собственных компьютерах, советую воспользоваться бесконечным циклом, описанным выше, а t следует отсчитывать в радианах.

Составляя программу MADNESS (безумие), читатели могут сделать ее обобщенной, расставив константы перед четырьмя тригонометрическими функциями. Тем самым перед ними откроется практически бесконечное пространство для исследований.

Программа CURVES, вдохновившая Миллера, является довольно универсальной. Она была создана компанией Bridge Software. Обладающая некоторыми полезными дополнительными возможностями и очень интересными функциями, она может успешно сыграть роль невидимого профессора. Но эта компания продает также и другую программу под названием SURE, «продукт» которой показан внизу на с. 82. Пользователи этой программы указывают функцию двух переменных и затем наблюдают, как программа рисует трехмерные пейзажи с горами и долинами. Обе программы относительно недороги.

Bridge Software — это не единственная маленькая компания, занимающаяся производством подобного математического обеспечения. Для любителей комплексных функций компания Lascaux Graphics создала программу под названием $f(z)$. Наверное, многие из вас помнят множество Мандельброта, о котором последний раз речь шла в январском номере журнала. Но, может быть, немногие читатели помнят, что все действие развивалось там на комплексной плоскости. Функции, которые строит программа фирмы Lascaux Graphics, отображают комплексную плоскость саму на себя. Мы не можем здесь наблюдать четырехмерного пространства, однако мы можем почти физически ощущать комплексную функцию по мере того, как на экране проявляется действие, оказываемое ею на комплексную плоскость: регулярную плоскую полярную решетку (см. рисунок внизу) комплексная функция превращает в грациозную криволинейную структуру, которую можно уподобить картине течений в математическом океане. Хотя $f(z)$ стоит несколько дороже программы CURVES, она может доставить много удовольст-



Комплексная функция (слева) *и ее логарифм* (справа)

WWW.SCIAM-MAGAZINE.NAROD.RU

вия читателям, которые хотят быть посвященными в самые глубокие тайны, присущие графическому представлению функций на экране дисплея.

НАНОТЕХНОЛОГИЯ, статья о которой была опубликована в мартовском номере, вызвала оживленные отклики как со стороны скептиков, так и со стороны энтузиастов. Любопытный образчик скепсиса содержится в письме, присланном «Ассоциацией гуманистов Оклахомы» и подписанном К. Л. Уайлзом из Оклахома-Сити. Уайлз не верит в обещание вечной жизни, которое некоторые энтузиасты усматривают в нанотехнологии. В частности, он считает, что подводный нанокорабль, описанный мною с таким восторгом, во-первых, вряд ли когда-нибудь удастся создать, а если он и будет создан, то уж никак, по мнению Уайлза, не сможет обеспечить вечной жизни. Он пишет: «Будучи в достаточной мере скептически настроенными по отношению к любым сверхестественным бессмертным существам, мы полагаем, что тем, кто рассчитывает на успехи нанотехнологии, придется ждать очень долго... Даже если подводные нанолодки в кровеносных сосудах — это безбожная идея, мы все же должны считать утверждение о ее реализуемости совершенно невероятным и требующим невероятного доказательства, — оно почти так же невероятно, как утверждение о том, что возможна сверхестественная вечная жизнь». Хотелось бы услышать мнение других читателей на этот счет.

Э. Дрекслер, конечно, находится в лагере защитников нанотехнологии. Поступая вполне ответственно, он тщательно избегает любых утверждений, выходящих за рамки возможности максимального продолжения человеческой жизни, которого можно достичь при помощи его «исцеляющих машин». Писатель П. Прус из Сан-Франциско, работающий в жанре научной фантастики, в восторге от идей Дрекслера. Он изучил некоторые возможности, которые, как ему кажется, несет в себе нанотехнология. В последней книге Пруса «Человеческая ошибка» описана гипотетическая самовоспроизводящаяся наномашина, которую случайно проглотил талантливый, но не очень приятный в жизни исследователь, работающий в одной из биотехнологических компаний Западного побережья США. После этого исследователь претерпевает удивительные превращения и даже обретает способность любить.

Наука и общество

СОИ: шаг вперед

СОЗДАНИЕ научно-исследовательской базы, от которой зависит будущее стратегической оборонной инициативы (СОИ), только началось. Организация так называемого Национального испытательного центра отвечает масштабам и целям программы СОИ: он обосновался под прикрытием фирмы Rockies на территории военно-воздушной базы «Falcon» близ Колорадо-Спрингс, а его штат будет насчитывать 2000 человек, которые разместятся в оффисе площадью свыше 30 тыс. кв. м. Центр будет «сердцем» общенациональной компьютерной сети под названием Национальный испытательный полигон. Компания Martin Marietta Corporation получила контракт на сумму 508 млн. долл., по которому в течение 5 лет она обязана построить и сдать в эксплуатацию компьютерную сеть; общая сумма расходов на создание полигона составит, по-видимому, около 1 млрд. долл. Официальная «закладка» центра состоялась в марте, но исследования в нем уже проводятся, пока во временных помещениях. Какова цель создания всего этого комплекса? Имитация реагирования системы C^3 (от трех английских слов: command, control и communication — командование, управление и связь. — *Ред.*) в случае массированного ядерного удара.

Обеспечение эффективного функционирования подсистем командования, управления и связи в составе многоуровневой системы обороны, включающей в себя сотни боевых средств наземного и космического базирования, представляет собой невероятно сложную задачу даже для крупнейших специалистов, привлеченных к разработке СОИ. Эта система должна анализировать данные, поступающие от огромного количества датчиков, предлагать военному командованию на выбор ряд альтернативных ответных действий, вести слежение за тысячами боевых головок и уметь отличать их от ложных целей, выбирать нужное средство для поражения целей и сигнализировать об их уничтожении. И все это за считанные минуты.

Такую систему ни при каких обстоятельствах нельзя испытать в реальных условиях. Поэтому ее создатели всю надежду возлагают на моделирование. Национальный испытательный полигон будет одной из наиболее мощных когда-либо созданных моделирующих систем. Ключевым звеном в системе будут два суперкомпьютера типа «Cray» и две большие вычислительные машины фирмы IBM; к ним предполагается подключить множество менее мощных компьютеров и экспериментальных процессоров с параллельным режимом обработки данных. Соединенный с основными вычислительными центрами, рассредоточенными по всей стране, Национальный испытательный полигон будет моделировать различные ситуации военного нападения и ответные действия оборонного характера. Он также сможет моделировать собственные сбои в системе C^3 и воздействие поражающих факторов ядерного взрыва на датчики.

Как заявил ответственный за программное обеспечение Национального испытательного полигона полковник Т. Лейб, в настоящее время система способна моделировать лишь небольшую часть крупномасштабного нападения, и то с невысокой степенью точности. Ответственная за управление боевыми операциями, разрабатываемыми системой C^3, В. Коблер из командования стратегической обороной сказала, что моделирование на испытательном полигоне более сложных ситуаций позволит проверить качество функционирования системы C^3 при ее рассредоточенной архитектуре. Систему в принципе можно сделать такой, что сбои в одном или нескольких узлах не повлияют на ее работоспособность.

В какой же степени можно доверять результатам моделирования на этой системе? Как указывает Р. Гарвин из Исследовательского центра Томаса Уотсона фирмы IBM, имитаторы полета самолетов, например, можно назвать реалистичными, потому что о поведении самолетов в воздухе мы многое знаем из накопленного практического опыта; аналогичного опыта в отношении ядерных взрывов в воздухе у нас нет.

При разработке средств обороны часто возникает сложнейшая проблема, связанная с точным определением характеристик средств нападения. По словам Д. Уэйсса из корпорации Software Productivity Consortium в Рестоне (шт. Виргиния), который занимался изучением программного обеспечения для систем ракетной обороны по заданию Управления по научно-технической оценке, любые изменения характеристик тех или иных видов вооружения требуют серьезных изменений в программном обеспечении процесса моделирования. Дж. Шор из компании Entropic